IAIK/Stiftung SIC

http://jce.iaik.tugraz.at

IAIK iSaSiLk

iSaSiLk is an implementation of the SSLv2 (client-side), SSLv3, TLS 1.0, TLS 1.1, TLS 1.2 and TLS 1.3 protocols written in the Java^{TM} language. It supports all standard cipher suites (except for Fortezza), including AES, GCM and PSK cipher suites. It can be used to develop secure client/server applications for communicating with any SSL/TLS compliant server (like Mircosoft IIS and Apache), or client (like Microsoft Internet Explorer, Mozilla Firefox, Google Chrome or Opera), respectively. iSaSiLk implements the standard TLS extensions and is highly configurable. It operates on top of the IAIK-JCE crypto toolkit but will work with any alternative JCE implementation and allows easy integration of smartcards or other hardware security modules.

<u>Main Benefits</u>

- Simplicity: iSaSiLk is very easy to use. Just a few number of lines of codes are necessary for turning a common client/server application into a secure one, e.g:

  ```
  SSLClientContext context = new SSLClientContext();
  SSLSocket s = new SSLSocket(host, port, context);
  PrintWriter out = new PrintWriter(s.getOutputStream());
  BufferedReader in =
    new BufferedReader(new InputStreamReader(s.getInputStream()));
  out.println("GET / HTTP/1.0");
  out.println();
  out.flush();
  String line;
  while ((line = in.readLine()) != null) {
    System.out.println(line);
  }
  ```

  Customizing and configuring an iSaSiLk application according to specific requirements is quite uncomplicated and can be learned rapidly with help of the documentation and demo samples included in the iSaSiLk distribution.
- Performance: Because of its speed optimized design iSaSiLk also is the ideal choice for heavy-loaded server applications.
- Reliability and Interoperability: As one of the world-first Java^{TM} SSL libraries, the initial version of iSaSiLk has been released already in

1997. Since then iSaSiLk is continuously maintained and updated to keep track with new standard versions and, if required, implement countermeasures to defend from attacks against the SSL/TLS protocol. Throughout its lifetime iSaSiLk has been and is successfully used in many applications all around the world.

- Configurability and Extensibility: iSaSiLk is highly configurable and can be easily extended for using alternative crypto providers, certificate handling strategies or, for instance, session management techniques. The cipher suite and compression method framework allows it to implement and plug-in private cipher suites and compression methods. Although most commonly iSaSiLk will be used with Java^{TM} Sockets and URL handlers, the transport-neutral design makes it possible to run SSL/TLS over any pair of streams.

Feature List

- Implemented entirely in the Java™ language guaranteeing cross platform portability
- Works on JDK Versions 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12 and compatible (JDK 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12 are also called JDK 5, 6, 7, 8, 9, 10, 11, 12 respectively)
- Mature product with a proven 15 year track record in real world applications
- Centralized security policy configuration
- Uses Socket API to allow easy upgrade of existing network applications
- Support for the HTTPS protocol via the standard JDK URL framework
- Secures Java™ RMI calls
- Supports client side SOCKS and HTTPS proxies
- Multithreading safe
- Client and server implementation of SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2 and TLS 1.3; client implementation of SSL 2.0
- Ensures that the most secure configured protocol version and encryption methods shared by client and server are used
- Supports all standard defined cryptographic algorithms including RSA, DSA, Diffie-Hellman, AES, Triple DES, DES, IDEA, RC2, RC4™, MD5, and SHA-1, SHA-2.
- Supports AES Galois Counter Mode (GCM) Cipher Suites for TLS according to RFC 5288
- Supports ECC cipher suites according to RFC 4492 (Named Curves)

- Supports all TLS defined NIST (RFC 4492) and Brainpool (RFC 7027) curves
- Supports x25519 and x448 ECDHE key exchange, and Ed25519 and Ed448 EdDSA signatures (RFC 8422; experimental)
- Supports RSA-PSS signature algorithms (rsa_pss_rsae_sha256, rsa_pss_rsae_sha384, rsa_pss_rsae_sha512, rsa_pss_pss_sha256, rsa_pss_pss_sha384, rsa_pss_pss_sha512)
- Supports ECC Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM) according to RFC 5289
- Supports Camellia cipher suites according to RFC 4132
- Supports Camellia Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), including PSK suites, according to RFC 5932/6367
- Supports all Pre-Shared key (PSK) cipher suites defined by RFC 4279, RFC 4785
- Supports Pre-Shared Key Cipher Suites with SHA-256/384 and AES Galois Counter Mode according to RFC 5487
- Supports Pre-Shared Key ECDHE_PSK Cipher Suites according to RFC 5489
- Easy plug-in of private cipher suites
- Public key server authentication, optional client authentication (ECC, RSA, DSA, and Diffie-Hellman) or fully anonymous connections
- Supports all standard TLS extensions defined by RFC 4366
- Supports Session Resumption without Server-Side State (SessionTicket extension according to RFC 4507 and RFC 4507bis)
- Supports the *extended_master_secret* extension as specified in RFC 7627 to calculate the master secret in a way that cryptographically binds it to important session parameters
- Supports tls-unique and tls-server-end-point channel bindings (RFC 5929)
- Session caching for high performance connection establishment
- Security parameter renegotiation on demand
- Application Extensible Design
  - Can perform SSL/TLS over any pair of streams and over an existing socket
  - Pluggable custom certification path verification
  - Pluggable custom session management
  - Allows private application defined encryption methods
  - Allows private application defined compression functions

- Proven Interoperability
    - Interoperates with any SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2 and TLS 1.3 implementation
    - Interoperability tested among others with clients Mozilla, Firefox, Microsoft Internet Explorer, Opera, Google Chrome.
    - Interoperability tested with servers from Microsoft, Oracle, IBM, Apache (SSLeay, OpenSSL) and others.

- Cryptographic Provider Independence
    - Can be used with any JCA/JCE 1.2 (or later) compliant cryptography provider
    - Can use several different cryptography providers at the same time
    - Easy integration of Smartcards and other secure hardware devices
    - Comes with the IAIK JCE provider by default (included in the license)